N

PTO 04-2921

Japanese Kokai Patent Application
No. Hei 7[1995]-93137

AUTOMOBILE CONTROL DEVICE AND PROGRAM DEVELOPMENT DEVICE

Tsuneo Kanokohata

UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. APRIL 2004
TRANSLATED BY THE RALPH MCELROY TRANSLATION COMPANY

JAPANESE PATENT OFFICE
PATENT JOURNAL (A)
KOKAI PATENT APPLICATION NO. HEI 7[1995]-93137

Int. Cl.^{6}:	G 06 F 9/06
Sequence No. for Office Use:	9367-5B
Filing No.:	Hei 6[1994]-61072
Filing Date:	September 18, 1989
Division of:	Hei 1[1989]-240082
Publication Date:	April 7, 1995
No. of Claims:	2 (Total of 11 pages; OL)
Examination Request:	Filed

AUTOMOBILE CONTROL DEVICE AND PROGRAM DEVELOPMENT DEVICE

[Jidosha seigyo sochi to puroguramu kaihatsu sochi]

Inventor:	Tsuneo Kanokohata
Applicant:	000005108 Hitachi, Ltd.

[There are no amendments to this patent.]

Claims /2*

1. An automobile control device, characterized by the fact that an automobile control device that applies arithmetic processing of a prescribed control program to detected signals introduced from various kinds of sensors and that outputs the processed result as a control signal to an object being controlled is equipped with a one-chip microcomputer built with a memory for storing the above-mentioned control program, which has been prepared in a high-level language, in a form translated into a machine language.

---

* [Numbers in the margin represent pagination in the original text.]

2. A program development device, characterized by the fact that a program development device for vehicles that is operated by a DC power source for an automobile and that develops the control program of Claim 1 is equipped with a bus that is connected via a connector to a bus of a control device which is loaded with a one-chip microcomputer for control and controls vehicle equipment, and a microcomputer that translates the control program, which is connected to said bus and prepared in a high-level language, into a machine language, and a real-time multitask OS that is an OS with a specification common to an OS of the above-mentioned one-chip microcomputer, corrects the above-mentioned control program, retranslates it into the machine language, and implements it by the above-mentioned control program, while introducing detected signals from various kinds of sensors connected to the above-mentioned control device during traveling of a vehicle and while controlling the above-mentioned vehicle equipment by arithmetic processing through the above-mentioned control program.

Detailed explanation of the invention

[0001]

Industrial application field

The present invention pertains to an automobile control device and a program development device.

[0002]

Prior art

A microcomputer (hereinafter, abbreviated to micon) for automobiles is used in various kinds of automobile control devices such as for engine ignition and fuel control, air conditioner control, transmission control, active suspension control, or brake control, and it is normal to prepare a one-chip microcomputer for each control device. As such a one-chip microcomputer, for example, there is a microcomputer for controlling an air conditioner shown in Japanese Kokai Patent Application No. Sho 63[1988]-11169, and its control program is prepared in accordance with electronic circuits designed for control. The control program of such a one-chip microcomputer for automobiles is usually described in an assembly language (or a machine language), and the reasons for this are mainly the following two.

(1) The limitation of memory (ROM, RAM) capacity is strict.

(2) A fast processing speed is required.

[0003]

Here, in condition (1), since cost reduction of automobile parts is very hard to achieve, a 4-bit or 8-bit one-chip microcomputer favorable in terms of cost has been used as the microcomputer. On the other hand, in a one-chip microcomputer, due to the limitation of its degree of integration, only a slight capacity for both ROM and RAM can be built in it, and in order to meet necessary control specifications in them, programs must be described in an assembly language for which the programs are most compact and which can also be effectively utilized in RAM. If ROM and RAM of another chip are externally installed to escape from the limitation of memory capacity, space is required for the IC for address data, wiring patterns on a printed-circuit board for the connection between each chip in addition to the ROM and RAM being installed are required, so that the number of parts is increased, the control circuit is enlarge, and the cost is significantly raised.

[0004]

On the other hand, in condition (2), since a conventional microcomputer has a relatively slow processing speed, the time required for one round of a main program is lengthened with increase of the number of steps of the program, so that the responsivity of a control system is gradually decreased. In practice, it is not rare to require a time of about 0.5 sec for one round of the main program, though this is largely influenced by the constitutional method of an interrupt handling program. Therefore, it is also in demand that the number of steps required for one round of the program be very small to improve the responsivity of a control system, and for this condition, an assembly language in which programs are most compact and a fast processing speed can be obtained must be used.

[0005]

Problems to be solved by the invention

Along with the recent progress of manufacturing technology of integrated circuits, the capacity of RAM and ROM built in a one-chip microcomputer has tended to increase. However, unlike microcomputers used in home electric appliances, in the one-chip microcomputer used as an automobile controller, the number of tasks processed by the one-chip microcomputer has increased, so that several kinds of control programs must be loaded in a memory with increased capacity. For this reason, it is also in demand that the control programs be stored without problems in a memory whose memory is increased. Furthermore, the automobile control device periodically implements tasks that must finish processing within a determined time of several ms or several tens of ms and output the arithmetic result, so high-speed processing is also in demand.

[0006]

In assembly language, since the program being implemented is compact, the memory capacity can be reduced and high-speed processing can be realized, so the automobile control device can simultaneously respond to the above-mentioned two demands. On the other hand, in the preparation of assembly language, since the hardware operation is described one step by one step, the programming work is complicated, so that the meaning of the program is difficult to understand and the number of lines of the program is increased. For this reason, there are the following problems.

[0007]

(1) Unless a small number of programmers handles the assembly language, preparation and change of control programs is difficult, and a long time is also required for the verification of programs.

[0008]

(2) Since assembly language has no compatibility with a microcomputer of a different architecture, programs must be re-prepared when the kind of microcomputer is changed, the correction and change of programs also require a facility of the same architecture as that of the microcomputer being used, and an immediate program change during a laboratory or outdoor running test is difficult.

[0009]

(3) In the experimentation of operation confirmation, etc., recording of data in the microcomputer or incorporating of display functions to the outside are difficult due to the /3 limitation of the memory capacity, and verification of the control responsivity and decision work of the specification are difficult.

[0010]

(4) In the program development device and the final microcomputer product, the architectures are generally different, and it is necessary to partially change programs (especially, change of the part related to input and output processing), so that the workability and program reliability are lowered.

[0011]

The objective of the present invention is to provide an automobile control device, which allows easy preparation, change, and verification of control programs even by a general programmer who is not skilled in handling assembly language, does not require change in control programs even if the kind of microcomputer is changed, and allows easy change of programs at an experimental site, and its program development device.

[0012]

Means to solve the problems

The above-mentioned objective is achieved by describing a control program in a high-level language, translating the control program into a machine language, and storing it in a memory built in a one-chip microcomputer used in an automobile control device.

[0013]

Also, the present invention is achieved by using a specification common to operating systems (hereinafter, abbreviated to OS) of the program development device and the control program and by being able to directly control equipment by running the control program on the development device.

[0014]

Operation of the invention

If the control program is described in a high-level language instead of assembly language, since the contents of the high-level language are easily understood and the programming is also simple, the control program can be easily prepared and changed without a specific programmer, and the number of program lines is reduced by several lines or several tens of lines, compared with assembly language, so that development, maintenance, and verification of the program is easy.

[0015]

With the use of a common specification for the OS of the program development device and of the OS of the microcomputer, the operational environment of the control program is the same on the development device and on the microcomputer, and a control program verified on the development device can be transferred to the control device without any change. Therefore, work for correcting part of the program in accordance with the control device is not required, which is useful for reduction of work time and prevention of generation new mistakes. Also, since an actual equipment item can be directly controlled by the program in the program development device, the input and output functions of a large-capacity RAM or other storage devices of the program development device and external equipment can be utilized, and using these functions, arbitrary data are recorded and analyzed, and experimental programs on vehicles can be easily changed.

[0016]

Furthermore, with the use of system call to the OS in the input and output processing of the control program, since the difference in address between the development device and the kind of microcomputer and the difference in the processing sequence of the input and output parts are covered and hidden in the OS, they are not seen from the control program, so that the same control program can be used, regardless of the constitution of the kind of microcomputer, the part being used, and the development device.

[0017]

Application example

Next, an application example of the present invention is explained referring to the figures. Figure 2 shows the entire constitution of an air conditioner. Air absorbed from an external air suction port 4 or internal air suction port 5 in accordance with the degree of opening of an intake door 10 is sent to an evaporator 2 by a blower motor 1 and cooled. Next, in accordance with the degree of opening of an upper air mixing door 11 and a lower air mixing door 12, cool air bypassing a heater core 3 and air reheated by passing through the heater core 3 are mixed downstream, blown as mild air into a vehicle cab from differential blow-off port 6 (to a front glass), vent blow-off port 7 (to the upper body of passengers), and floor blow-off port 8 (to the feet of passengers) in accordance with the degree of opening of a floor door 13 and a vent door 14 for switching the blow port, so that the temperature in the vehicle cab is controlled.

[0018]

A control device 20 has a differential duct sensor 21 for detecting the blow-off air temperature of the differential blow-off port 6, a vent duct sensor 22 for detecting the blow-off air temperature of the vent blow-off port 7, a floor duct sensor 23 for detecting the blow-off air temperature of the floor blow-off port 8, an external air temperature sensor 24 for detecting the air temperature outside the vehicle cab, a vehicle cab temperature sensor 25 for detecting the air temperature in the vehicle cab, and a control panel 26 that inputs a detected signal from a sunlight sensor 27 for detecting the strength of sunlight, controls electromotive actuators 15-19 for adjusting the degree of opening of each door 10-14 and controls the blower motor 1, and said device is equipped with switches for applying various kinds of air conditioner operations by passengers and a display device for notifying the operation state to the passengers.

[0019]

Figure 3 shows a further detailed constitution of the control device 20, and the same symbols as those of Figure 2 show the same parts as those of Figure 2. In addition, a compressor relay 40 for turning on and off a compressor and a negative pressure value 41 for controlling opening and closing of the flow of an engine coolant are provided as loads of the control device 20, though they are not shown in Figure 2.

[0020]

Processors, ROM, RAM, I/O port, A/D converter, etc., which are not shown in the figure, are built in a one-chip microcomputer 30 in the control device 20. A driver 31 controls the rotation (forward rotation, backward rotation, and stop) of the electromotive actuators, a fan control circuit 32 controls the applied voltage of the blower motor 1 for blowing air, and a solenoid driver 33 controls ON and OFF of the compressor relay 40 and the negative pressure value 41 as external solenoids.

[0021]

A ROM 38 for storing programs and control constants and a RAM 39 for storing data are connected with each other inside the control device 20 and are operated based on instructions of the one-chip microcomputer 30. If the ROM and the RAM in the microcomputer 30 have sufficient memory capacity, they [38,39] are not required.

[0022]

A signal from an operation switch 35 on the control panel 26 assembled into the control device 20 is input into the microcomputer 30 via an interface circuit 34, and the current air /4
conditioner operation state, the set target temperature, etc., are output to an indicator 37 via an indicator driver 36 and are indicated to the passengers.

[0023]

Figure 4 is a flow chart showing a control program example implemented by microcomputer 30. First, at step 500, a register and a RAM in the microcomputer are initially set, and an input and output environment for an air conditioner system is arranged. Next, control steps 502-511 of the actual air conditioner system are repeated by the repetitive processing of step 501 until the power is cut off. In other words, at step 502, temperature signals input from various kinds of sensors 21-25, 27, etc., are read into the one-chip microcomputer 30, and at step 503, data such as temperature signals read at the previous step are converted into internal data by correction of the nonlinearity of the sensors, the conversion of units, etc. Then, at step 504, the control target temperature Tso as a target cab temperature is calculated. In the calculation, the set temperature selected by passengers is corrected in accordance with the external air temperature and the operation mode so that a pleasant temperature space can be maintained without an adjusting operation of the passengers. At the next step 505, starting with the difference between the above-mentioned control target temperature Tso and the actual vehicle cab temperature Tr detected by the vehicle cab temperature sensor 25, the blow-off target temperature Tdo required for maintaining a pleasant temperature space in accordance with the values of the external air temperature Ta and the amount Zm of sunlight is calculated. Next, at step 506, the degree of opening of the air mixing (A/M) doors 11 and 12 is calculated. Here, this degree of opening of the air mixing doors 11 and 12 is that required for the difference between the target blow-off temperature Tdo calculated at the previous step and the actual blow-off temperature Td detected by the duct sensors 21, 22, and 23 corresponding to each blow-off mode to approach 0.

[0024]

In this manner, the calculation of the temperature control relation is finished, and a blower motor control target voltage for determining the amount of blow-off air is calculated at the next step 507. Here, the voltage is calculated so that a low amount of air is obtained when the difference between the control target temperature Tso and the actual vehicle cab temperature Tr is almost 0, and the voltage is calculated so that the amount of air may be increased with increase of the difference. At the step 508, an optimum distribution of the amount of air of each blow-off 6-8 is determined in accordance with the external air temperature Ta, the amount Zm of sunlight, the blow-off temperature Td, etc., and an internal circulation air corresponding to thermal load is determined. At the next step 509, ON and OFF of the compressor are determined. In general, it is determined that the compressor should be forcibly turned off under a low external air temperature condition in which the thermal load is decreased (operation of relay 40). Also, at step 510, opening and closing (opening and closing the negative pressure valve) of a water cock is determined. Finally, at step 511, control output signals are simultaneously output to each actuator calculated in this manner, and the degree of opening of the air mixing doors, the blower motor applied voltage, the door state of the suction port and the blow-off port, ON and OFF of the compressor and the water cock, and the indicators on the control panel are controlled. With repetitive implementation of the above steps 502-511, the inside of the vehicle cab is maintained in a pleasant state in accordance with the conditions set by passengers.

[0025]

Here, using steps 505 and 509 of Figure 4 as an example, an example of a program characteristic of the present invention is explained. First, an example of a calculation equation of the target blow-off temperature Tdo at step 505 is shown.

[0026]

(Expression 1)

$$\begin{aligned} Tdo &= f(Ta) - Kz \cdot Zm + Kr\,(Tso - Tr) \\ &= 40 - Ka \cdot Ta - Kz \cdot Zm + Kr(Tso - Tr) \qquad (1) \end{aligned}$$

Where, $f(Ta) = 40 - Ka \cdot Ta$: Reference blow-off temperature

Ta: External air temperature

Zm: Amount of sunlight

Tso: Control target cab temperature

Tr: Vehicle cab temperature

Ka, Kz, Kr: Control constant

The reference blow-off temperature f(Ta) is a blow-off temperature such that passengers feel pleasant when there is no sunlight and the vehicle cab temperature Tr arrives at the control target cab temperature Tso and has a characteristic wherein it is almost linearly lowered with rise of the external air temperature Ta as shown in Figure 5.

[0027]

If the arithmetic of the equation (1) is programmed in a conventional assembly language, first, storage areas corresponding to each program variable are defined in the RAM. In the control of an air conditioner for automobiles, a storage area of about 2 bytes for one variable is required by the requirement of the temperature change range and its resolution, and for this reason, as shown in Figure 6(b), storage areas of 2 bytes are defined for each of the external air temperature Ta, the amount Zm of sunlight, the vehicle cab temperature Tr, the control target cab temperature Tso, and the target blow-off temperature Tdo. Since the values of the variables except for the calculated temperature Tdo have already been calculated at steps 503 and 504 and the values are stored in defined areas, the calculation of equation (1) is implemented using these values by a flow chart shown in Figure 6(a). In other words, at step 600, the reference blow-off temperature $40\text{-}Ka \cdot Ta$ is calculated, and the result is temporarily stored in the defined region of Tdo. Here, A→ B of the figure means that A is attained and substituted for B. Next, at step 601, amount $Kz \cdot Zm$ of correction of the blow-off target temperature for the amount Zm of sunlight is subtracted from the value of Tdo attained at step 600, and the result is re-stored in the area of Tdo. Next, in step 602, amount $Kr(Tso - Tr)$ of correction of the blow-off target temperature obtained based on cab temperature deviation is added to Tdo derived in step 601. The result is then stored in the area of Tdo. With the implementation of the above sequence, the target blow-off temperature data Tdo calculated by the above-mentioned equation (1) is obtained in the defined area of Tdo. On the other hand, at each of these steps, multiplication of data with a length of 2 bytes is required, and in a conventional 8-bit microcomputer, a subroutine for a double length multiplication is required at each step. Since the double length multiplication method is well known, its details are omitted, and a quadruple multiplication of data with a length of 1 byte and an addition by appropriately fitting these results to digits are required. Also, since the multiplication result has a length of 4 bytes, a memory area fitting it is prepared.

[0028]

An example of a program listing that describes the processing of the above equation (1) in assembly language is shown in Figures 7(a) and (b). (a) is the program of Figure 6(a) for calculating equation (1), and (b) is a subroutine with a 2-byte length multiplication used in it. This listing is an example for an 8-bit one-chip microcomputer HD6801 made by Hitachi, Ltd., and consists of about 50 digits. Also, the contents of the program are difficult to understand simply by seeing the listing, and much time is required for changing part of the arithmetic specification without a misunderstanding. Furthermore, since the microcomputer HD6801 shown in this example is equipped with addition and subtraction functions of 2-byte data, the program can be relatively simply assembled; however in the HD6805 made by the same company and frequently used for controlling an air conditioner for automobiles, since there are no addition and subtraction functions of 2-byte data, the program is further long and complicated.

[0029]

In the present invention, in order to solve the problems of such assembly language, the arithmetic of equation (1) is described in a high-level language. Figure 1 shows an example in which equation (1) is described in C language that is frequently used as a language for controlling a microcomputer. As seen from the figure, the program is very simple, and the description itself of the program is almost the same as that of equation (1), and the program preparation and the arithmetic specification change are very simple, compared with assembly language.

[0030]

Next, the comparison for the decision program of ON and OFF of the compressor of step 509 of Figure 4 is explained. Figure 8 shows an example of the ON and OFF decision characteristic of the compressor. As mentioned above, it is not necessary to operate the compressor at a low external air temperature, and for the purpose of preventing damage to the compressor due to compressor operation at a low thermal load, generally, when the external air temperature Ta is about 0°C or lower, the compressor is also forcibly turned off. Then, in order to prevent chattering at a switching point due to the influence of a fine variation of the signal from the external air temperature sensor, an ON and OFF decision with an installed hysteresis is made so that the compressor may be turned on at an external air temperature of #Hys (1°C in Figure 8) or more and the compressor may be turned off at an external air temperature of -#Hys or less.

[0031]

Figures 9 and 10 are flow charts for processing of step 509 of Figure 4. First, a subroutine of a size decision having a hysteresis characteristic (a subroutine having versatility) is explained by Figure 10. Figure 10(b) shows a Flag area in a memory, and 1 bit of the LSB is used as Bco-mp for storing the ON/OFF state of the compressor. Also, [IX] in Figure 10(a) means a data value stored in an index register IX, and it is a data value subjected to a size decision with hysteresis, that is, of the external air temperature Ta. At step 900 of Figure 10(a), first, whether or not the current compressor is in the ON state is determined. If Bcomp = "1," that is, if the current compressor is in the ON state, [IX] is replaced with Ta+#Hys at step 901, and if Bcomp = "0," that is, if in the OFF state, [IX] is replaced with Ta-#Hys at step 902. Then, at step 903, whether [IX] is positive or negative is determined. When the decision is positive, if Bcomp = "0" up to now," it means Ta-#Hys> 0, and if Bcomp = "1" up to now, it means Ta+#Hys > 0. As seen from Figure 8, since the compressor is in the ON state in either case, Bcomp = "1" at step 904. On the contrary, if the decision of step 903 is negative, if Bcomp = "0" up to now, it means Ta-#Hys < 0, and if Bcomp = "1" up to now, it means Ta+#Hys < 0. Since the compressor is in the OFF state in either case, Bcomp = "0" at step 905.

[0032]

Figure 9 is a main program for implementing step 509 of Figure 4. At step 901 or 902 of Figure 10(a), since [IX] must be rewritten, it is necessary to set the value of the external air temperature data to the register IX each time the processing of Figure 10(a) is started. For this reason, as shown in Figure 9(b), a working area called Work is prepared apart from the area for storing Ta in the memory. Then, at step 800 of Figure 9(a), the value of the external air temperature Ta is copied in the Work area, and at step 801, after the address of the Work area, that is, #Work is substituted for the index register, the subroutine is called at step 802.

[0033]

Listings in which the above processing of Figures 9 and 10 are described in an assembly language are respectively shown in Figures 11(a) and (b). Compared with the above-mentioned Figures 7(a) and (b), the number of program steps is very small; however from these listings only, it is also very difficult to read out which processing is implemented by the program.

[0034]

Figure 12 shows a program example of the case where the above-mentioned decision processing 509 is described in C language, and the number of program lines is not reduced, compared with that of conventional Figures 11(a) and (b). However, the contents can be easily understood by any technician who handles C language, and the characteristics can also be easily changed.

[0035]

As explained in the application example of Figures 1 and 12, with the description of the control program in C language as a high-level language, the number of program description lines is reduced (especially, the numerical value arithmetic processing is reduced), and the contents are easily understood, so that the specification change of the program and the verification of the operation contents can be easy and reliable. Also, in a representative high-level language like C language, a good-quality compiler has already been provided to almost all common microcomputers, and even if the structure of each microcomputer is different, if each private compiler is loaded, the same control program can be used.

[0036]

On the other hand, an area for storing a translated object program (machine language) is required, and unavoidably, the object program is a command column with a rather poor /6 efficiency due to an automatic translation of the compiler, and the processing speed is usually lower than that of a program prepared in an assembly language. However, the increase of the memory capacity and the decrease of the processing speed, as mentioned above, can be sufficiently absorbed by the progress in hardware.

[0037]

Next, as another application example of the present invention, an example in which the arithmetic of calculation equation (1) and the ON and OFF decision of the process shown in Figure 8 are described in BASIC high-level language is shown in Figures 13 and 14. Compared with Figures 1 and 12 using C language, the range of humans who can prepare and change the program is further widened. The same effects as in the case when using C language can be realized. Also, since BASIC language has been widely used for PCs, more people can write and change programs using this language as compared to using C language. At the same time, the BASIC language is called an interpreter language, and even in a program implementation environment state, since the program can be temporarily stopped and easily changed and supplemented, this language can immediately respond to a specification correction during the development of a program specification.

[0038]

On the other hand, in the BASIC language, the processing speed is generally slow; however the processing speed problem is solved by the improvement of processibility of the microcomputer. BASIC programs built with a BASIC interpreter for directly implementing programs can also be reported, and the required ROM capacity and processing speed are further improved. In addition, BASIC includes an intermediate code (I code) interpreter type or a compiler type similar to C language, and using these types, the program product can be compacted, and program processing can be accelerated.

[0039]

Other high-level languages can also be similarly applied, and compared with those limited to automobile control and air conditioner control, a high-speed and compact private high-level language can be designed and used.

[0040]

Next, as another application example of the present invention, an application example in which a program development device and an air conditioner for automobiles are connected and the air conditioner is controlled by a program described in assembly language or a high-level language on the program development device is shown in Figure 15. Here, the program development device has functions such as editor that is used in the preparation and correction of the program, compiler for translating the prepared the high-level language program, assembler for translating the assembly language program, and debugger and emulator for check and evaluation and prepares and corrects the OS and control program for a one-chip microcomputer.

[0041]

The upper block of Figure 15 shows an internal constitutional example of the control device 20 of the air conditioner of Figure 2. The difference from Figure 3 is that the one-chip microcomputer 30 is removed from Figure 3, signal lines from an interface circuit and a driver circuit to each input and output terminal are drawn out via a bus line 43 instead of by it, and an interface 42 is installed in the input parts of each of sensors 23-25 and 27.

[0042]

On the other hand, the lower block of Figure 15 shows an internal constitutional example of a program development device 59 and consists of microcomputer 45, ROM 46, RAM 47, timer counter 48, storage devices 50 and 51 (floppy disk, root disk, optical disk, bubble memory, etc.), keyboard 53, display device 55, common measurer 57 connected to the outside, processor 58 (printer, plotter, data logger, EPROM writer, etc.), interface circuits 49, 52, and 54, common interface circuit 56 (RS-232C, etc.) for connecting the program development device and an external common device, and bus 60. The bus 60 and the bus 43 of the control device 20 are connected by a connector 44.

[0043]

As shown in the above constitution, a program development device 59 is loaded with a real-time multitask OS, is equipped with a large-capacity RAM 47 and large-capacity and high-speed storage devices 50 and 51, can operate an editor, various kinds of assemblers, complier of a high-level language starting with C language and BASIC language, interpreter, etc., required for program development on the real-time multitask OS, and can directly operate the control device 20 via the drawn-out data buses 60 and 43 from the microcomputer. In addition, the program development device 59 can be operated by 100 V AC or a DC power source for automobiles.

[0044]

Therefore, using this program development device, a control program for the air conditioner is prepared in assembly language or a high-level language, the program is run on the microcomputer 45, and the air conditioner of a vehicle can be directly controlled via the control device 20. In particular, in a test run of the pleasantness evaluation in an actual vehicle, this entire system is loaded in the vehicle, the air is controlled by the program development device 59, and readout data input from each sensor, arithmetic data in the control program, operation situations of each actuator, the temperature of each part of the vehicle read out via a common external device, pressure data, etc., are sequentially recorded in real time in the storage devices 50 and 51 such as disks and provided for post-analysis. At the same time, the change details of each data item are displayed in real time on the display device 55 and the measurer 57, and if necessary, the air conditioner control program can also be amended and recompiled during the vehicle run. Also, if the program development device 59 is used, control panel 26 accordingly integrated with the control device 20 is not required.

[0045]

Also, in the control device of Figure 3 as a final product, a common OS in which only functions unnecessary for the product are erased from the OS used in the program development device of this application example is loaded, and a control program of which the operation has been verified is obtained without other changes.

[0046]

As usable OSs of the program development device, there are OS9, I-TRON, μ-TRON, any real-time OS supplied by any microcomputer maker, etc., however a compact /7 high-performance OS for this system can also be developed and used.

[0047]

According to this application example, the control program development and specification change period can be largely shortened, and the quality of a given program can be largely improved.

[0048]

Next, an application example in which an input and output is realized by a system call operation on the OS without direct control from a program to input and output interface addresses in inputting and outputting signals or data is shown in Figure 16. It is a program example in which an arithmetic result is output from C language operating on the OS, and the control output of 1 byte containing Bcomp showing ON and OFF of the compressor calculated by the program of Figure 12 is output to an actual output port defined as Oport.

[0049]

According to this application example, even if the input and output port addresses or input and output sequences are changed by differences of the constitution of the control circuit or the parts used, since all these differences can be absorbed in the OS, the control program including the input and output processing programs shown in Figure 16 is not influenced. Therefore, even if the control program is used for other kinds of devices or if the control program prepared on the program development device shown in Figure 15 is applied to the device of Figure 3 with different input and output addresses, the control program can be used as is without any change, so that the development period of the control program can be greatly shortened and the quality of said program can be greatly improved.

[0050]

Various application examples have been shown for the control of an air conditioner for automobiles, however there are a number of applications of a one-chip microcomputer such as engine control. Needless to say, the present invention can also be utilized for them.

[0051]

Effect of the invention

According to the present invention, in a high-level language, since its program is easily prepared and the contents are easily understood, a control program can be easily prepared, changed, and verified by persons other than specially skilled programmers, and since the structure of a microcomputer is hidden by a compiler corresponding to the kind of device, the same control program can be used, even if the kind of microcomputer and the parts used are different. For these reasons, the number of development processes of the automobile control device and the manpower are reduced, so that the manufacturing cost can be reduced.

[0052]

Also, with the adoption of a common specification for the OS of the program development device and the control device, the control program can be amended and changed during experiments in the development period of products or in an arbitrary place on an outdoor vehicle, and the prepared control program can be used as is in the control device without large change. Also, arbitrary data in the microcomputer can be simply recorded and displayed during the development period of products, and these data can also be automatically analyzed (processing such as graphing). Furthermore, with the use of system call to the OS in the input and output processing of the control program, since differences in the addresses of the input and output can be absorbed by the OS, the compatibility of the control program can be further raised. Accordingly, along with the above-mentioned effects, the present invention can contribute greatly to a large reduction in the period of development of the control program and specification changes and improvement of the reliability of said program.

Brief description of the figures

Figure 1 shows an example of a control program characteristic of the present invention.

Figure 2 shows the entire constitution of an air conditioner for automobiles and its control device.

Figure 3 is a block diagram showing an internal constitutional example of the control device.

Figure 4 is a flow chart showing an outline of the control sequence of an air conditioner for automobiles.

Figure 5 shows a characteristic example of a reference blow-off temperature.

Figure 6 shows an arithmetic sequence of a target blow-off temperature.

Figure 7 shows an example in which the sequence shown in Figure 6 is programmed in assembly language.

Figure 8 shows an example of ON and OFF characteristics of the compressor.

Figure 9 is a flow chart showing a decision sequence of ON and OFF of the compressor.

Figure 10 is a flow chart showing a decision sequence of ON and OFF of the compressor.

Figure 11 shows an example in which the decision sequences shown in Figures 10 and 11[sic; 9 and 10] are programmed in assembly language.

Figure 12 shows an example in which the decision sequences shown in Figures 10 and 11[sic; 9 and 10] are programmed in C language.

Figure 13 shows an example in which the program corresponding to Figure 1 is shown in BASIC language.

Figure 14 shows an example in which the program corresponding to Figure 12 is shown in BASIC language.

Figure 15 shows an application example of program development device characteristic of the present invention.

Figure 16 shows an example of a program for the input and output by a sequence call of an OS.

Explanation of symbols:

20	Control device
31	One-chip microcomputer
43, 60	Bus
44	Connector
45	Microcomputer
46	ROM
47	RAM
59	Program development device

```
td =40.+ka*ta-ks  sm+kr*(ta-tr);
return(td );
```

Figure 1

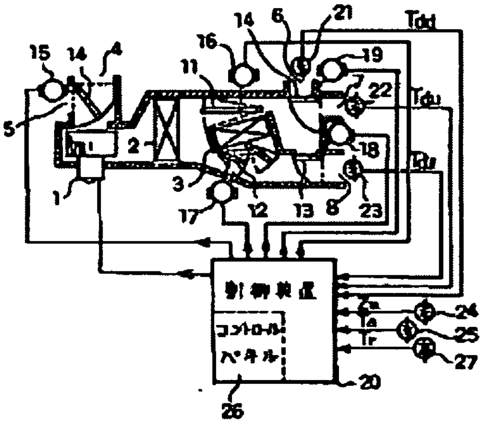

15
14
4
5
1
2
3
17
12
13
8
23
16
11
14
6
21
Tdd
19
7
22
Tdu
18
Tdf
制御装置
コントロール
パネル
Ts
Ta
Tr
24
25
27
20
26

Figure 2

Key:		
	20	Control device
	26	Control panel

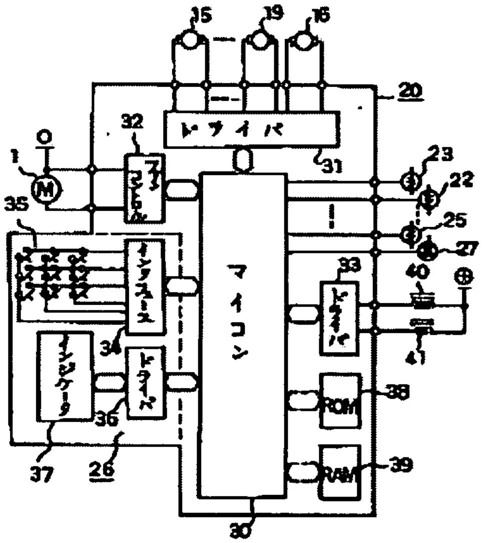

15
19
16
20
ドライバ
31
32
ファンコン
1
M
35
インタフェース
34
インジケータ
37
26
36
ドライバ
マイコン
33
ドライバ
23
22
25
27
40
41
ROM
38
RAM
39
30

Figure 3

Key:		
	30	Microcomputer
	31	Driver
	32	Fan controller
	33	Driver
	34	Interface

36	Driver
37	Indicator

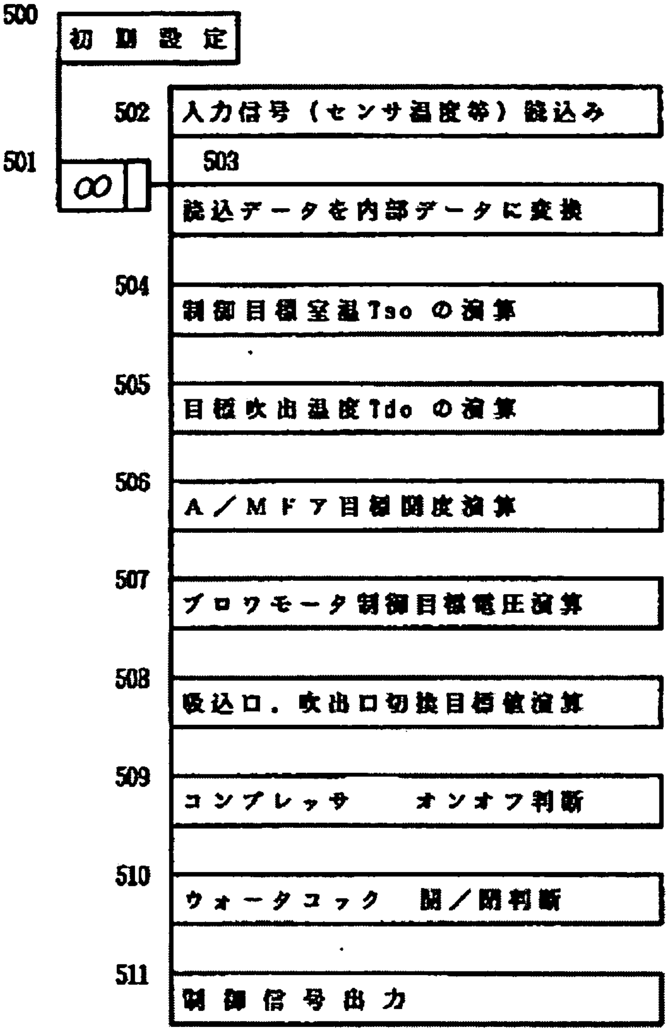

500
初期設定
502
入力信号（センサ温度等）読込み
501
∞
503
読込データを内部データに変換
504
制御目標室温Tso の演算
505
目標吹出温度Tdo の演算
506
A／Mドア目標開度演算
507
ブロワモータ制御目標電圧演算
508
吸込口．吹出口切換目標値演算
509
コンプレッサ　オンオフ判断
510
ウォータコック　開／閉判断
511
制御信号出力

Figure 4

Key:		
	500	Initial setup
	502	Reading of input signals (sensor temperature, etc.)
	503	Conversion of read data into internal data
	504	Calculation of control target cab temperature Tso
	505	Calculation of target blow-off temperature Tdo
	506	Calculation of a target degree of opening of A/M doors
	507	Calculation of blower motor control target voltage
	508	Calculation of suction port and blow-off port switching target value
	509	ON and OFF decision of compressor
	510	Opening/closing decision of water cock
	511	Control signal output

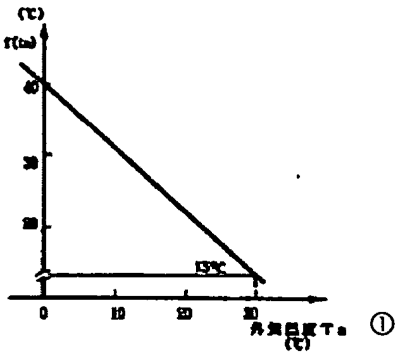


Figure 5

Key: 1 External air temperature Ta (°C)

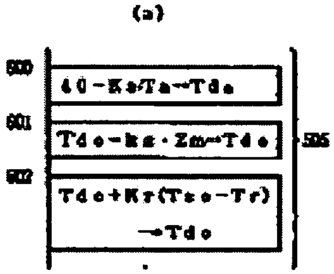

(a)
600
40−Ka·Ta→Tdo
601
Tdo−ks·Zm→Tdo
602
Tdo+Kr(Tso−Tr)
→Tdo
505

Figure 6

(a)

```
OprTdo  ldd  #ka
        std  Param1
        ldd  Ts
        std  Param2
        bsr  Mul10
        ldd  #40
        subd MulAns+1
        std  Tdo

        ldd  #kz
        std  Param1
        ldd  Zm
        std  Param2
        bsr  Mul10
        ldd  Tdo
        subd MulAns+1
        std  Tdo

        ldd  #kr
        std  Param1
        ldd  Ts
        std  Param2
        bsr  Mul10
        ldd  Tdo
        addd MulAns+1
        std  Tdo
        rts
```

(b)

```
Mul 10    clr  MulAns
          clr  MulAns+1

          lda  Param1+1
          ldb  Param2+1
          mul
          std  MulAns+2

          lda  Param1+1
          ldb  Param2
          bsr  MulSub10

          lda  Param1
          ldb  Param2+1
          bsr  MuiSub10

          lda  Param1
          ldb  Param2
          mul
          addd MulAns
          std  MulAns
          rts

Mulsub10  mul
          addd MulAns+1
          std  MulAns+1
          lda  #0
          adca MulAns
          sat  MulAns
          rts
```

Figure 7

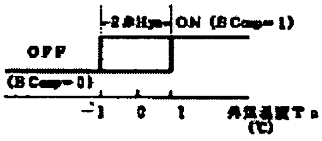


Figure 8

Key: 1 External air temperature Ta (°C)

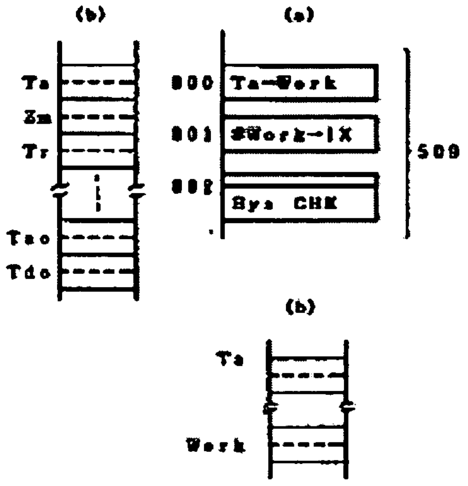

(b)
(a)
Ta
Xm
Tr
Tao
Tdo
800
Ta→Work
801
#Work→IX
802
Sys CHK
509
(b)
Ta
Work

Figure 9

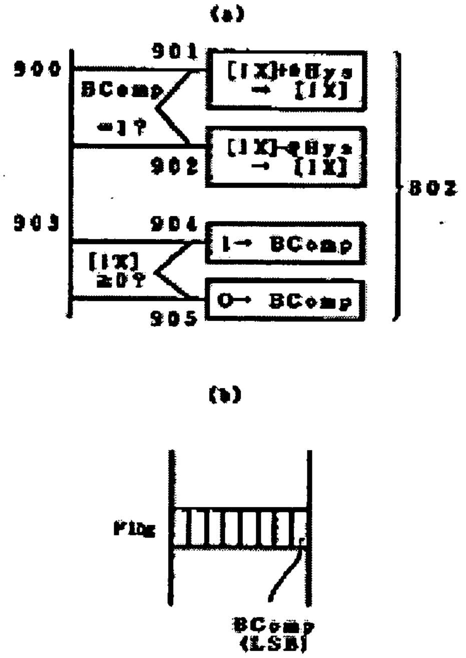

(a)
900
901
BComp =1?
[IX]+#Sys → [IX]
[IX]-#Sys → [IX]
902
802
903
904
1→ BComp
[IX] ≥0?
0→ BComp
905
(b)
Flag
BComp (LSB)

Figure 10

(a)

```
Contcomp  ldd  Ta
          std  Work
          ldx  #Work
          bsr  HysChk
          lda  Flag
          sta  OPort
          rts
```

(b)

```
*List 5

HysChk    lda  Flag
          bita #BComp
          bne  HysChk10
          ldd  0,X
          addd #Hys
          bra  HysChk20

HysChk10  ldd  0,X
          subd #Hys
HysChk20  bPl  HysChk30
          lda  Flag
          anda #^BComp
          bra  HysChk40

HysChk30  lda  Flag
          ora  #^BComp
HysChk40  sta  Flag
          rts
```

Figure 11

```
if (flag & bcomp)
     ta += Hys;
else
     ta -= Hys;
if ( ta )=0.)
      flag |=bcomp;
else
      flag &=bcomp;
```

Figure 12

```
Tdo=40,-Ka*Ta-Kz*Zm+Kr*(Ts-Tr)
```

Figure 13

```
IF  LAND (Flag, BComp)THEN
        Ta=Ta+Hys
ELSE
        Ta=Ta-Hys
ENDIF
IF  Ta>=0.  THEN
          Flag=LOR (Flag, BComp)
ELSE
          Flag=LAND (Flag, LNOT (BComp) )
ENDIF
```

Figure 14

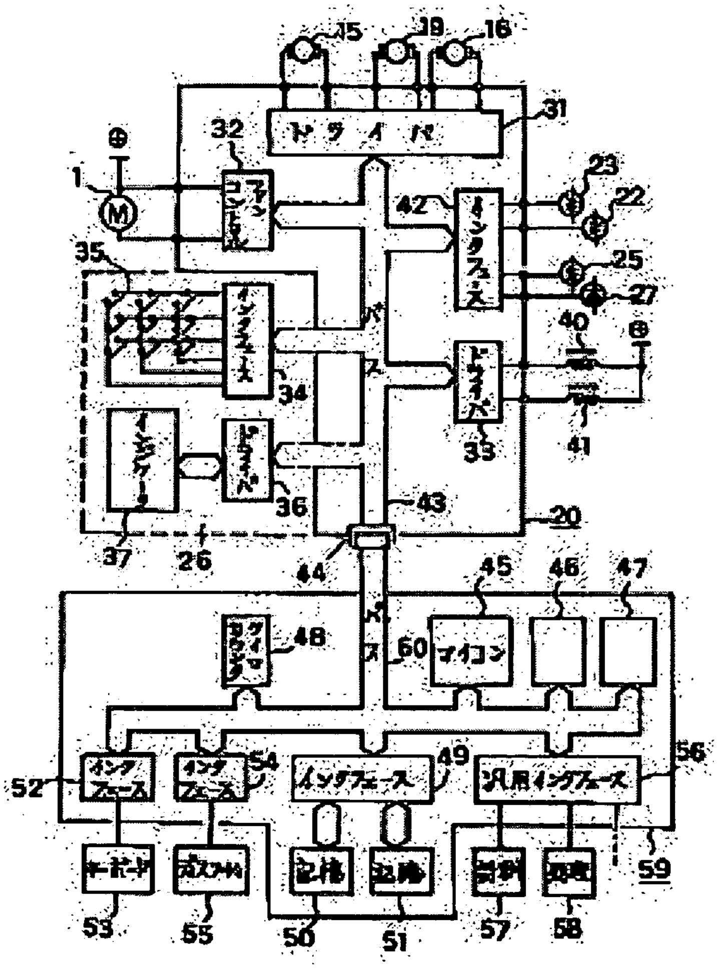
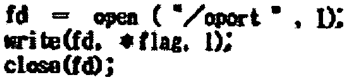

15
19
16
31
ドライバ
32
1
M
42
23
22
25
27
35
34
40
41
33
36
37
26
44
43
20
バス
45
マイコン
46
47
48
60
52
インタフェース
インタフェース
54
インタフェース
49
汎用インタフェース
56
キーボード
ディスプレイ
記憶
記憶
計測
通信
59
53
55
50
51
57
58

Figure 15

Key:		
	31	Driver
	32	Fan controller
	33	Driver
	34	Interface
	36	Driver
	37	Indicator

42	Interface
43	Bus
45	Microcomputer
48	Timer counter
49	Interface
50	Storage
51	Storage
52	Interface
53	Keyboard
54	Interface
55	Display
56	Common interface
57	Measurement
58	Processing
60	Bus



```
fd  =  open ( "/oport " , 1);
write(fd, *flag, 1);
close(fd);
```

Figure 16